МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ижевский государственный технический университет имени М. Т. Калашникова»

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Работа защищена с оценкой

«\_\_\_\_»

Дата \_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_/\_\_\_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к преддипломной практике на тему:

«Система управления учебным процессом. Серверная часть»

Выполнил:

Студент группы Б08-191-2 М.Л. Поздеев

Руководитель:

Старший преподаватель кафедры ПО К.С. Чернышев

Рецензия:

степень достижения поставленной цели работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

полнота разработки темы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

уровень самостоятельности работы обучающегося\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

недостатки работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[2. РАЗРАБОТКА ЗАДАЧ ПО СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ 3](#_Toc40998415)

[2.1. Разработка сервиса авторизации 3](#_Toc40998416)

[2.1.1. Описание постановки задачи 3](#_Toc40998417)

[2.1.2. Описание алгоритма регистрации пользователей 4](#_Toc40998418)

[2.1.3. Описание алгоритма аутентификации и авторизации пользователей 6](#_Toc40998419)

[2.2. Разработка сервиса ресурсов 10](#_Toc40998420)

[2.2.1. Описание постановки задачи 10](#_Toc40998421)

[2.2.2. Описание алгоритма предоставления данных для приложений преподавателя и студента 13](#_Toc40998422)

[2.3. Описание контрольного примера 18](#_Toc40998423)

[2.3.1. Назначение 18](#_Toc40998424)

[2.3.2. Исходные данные 18](#_Toc40998425)

[2.3.4. Результаты расчета 18](#_Toc40998426)

[2.3.5. Результаты испытания программы 20](#_Toc40998427)

2. РАЗРАБОТКА ЗАДАЧ ПО СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ

2.1. Разработка сервиса авторизации

2.1.1. Описание постановки задачи

2.1.1.1. Характеристика задачи

Задача – разработать сервис авторизации.

Сервис авторизации разрабатывается с целью обеспечения безопасности данных, хранимых в системе. Для реализации возможности масштабирования системы в будущем на большое количество университетов – было решено разработать собственный сервис авторизации, который будет предоставлять доступ мобильным приложениям пользователей к API системы.

Разработка сервиса авторизации связана с разработкой сервиса ресурсов системы. С помощью проверки доступа к методам API сервиса ресурсов – будет проверяться работоспособность сервера авторизации.

Разработка задачи будет завершена при успешном прохождении тестирования сервера.

2.1.1.2. Входная информация

Перечень входных данных при регистрации пользователей:

* номер зачетной книжки (в случае, если пользователь – студент);
* идентификатор преподавателя в системе (в случае, если пользователь – преподаватель);
* уникальный логин пользователя;
* пароль.

Перечень входных данных при аутентификации и авторизации пользователей:

* логин пользователя, созданный при регистрации;
* пароль пользователя, созданный при регистрации.

Перечень входных данных при валидации токена доступа:

* токен доступа.

Перечень входных данных при обновлении токена доступа:

* токен восстановления доступа;
* старый токен доступа.

Вся вышеперечисленная информация передается на сервер с помощью http запросов.

2.1.1.3. Выходная информация

Перечень выходных данных при регистрации пользователей:

* сообщение об успешной или о неудачной регистрации.

Перечень выходных данных при аутентификации и авторизации пользователей:

* токен доступа;
* токен восстановления доступа;
* идентификатор пользователя в системе.

Перечень выходных данных при валидации токена доступа:

* сообщение об успешной или неудачной валидации.

Перечень выходных данных при обновлении токена доступа:

* новый токен доступа при успешном обновлении токена;
* новый токен восстановления доступа при успешном обновлении токена;
* идентификатор пользователя в системе при успешном обновлении токена;
* ошибка обновления токена при не правильных входных данных.

Вся вышеперечисленная информация выдается сервером в ответ на http запросы.

2.1.2. Описание алгоритма регистрации пользователей

2.1.2.1. Назначение и характеристика

Регистрация необходима для создания пользователя в системе и его сопоставления с сущностью студента или преподавателя в базе данных.

Алгоритм должен сохранять данные пользователей в базе данных, при этом проверив существование этого пользователя как студента или преподавателя в системе. Пароли пользователей должны записываться в базу в виде хеша таким образом, чтобы одинаковые пароли имели разное представление в базе данных. Полученный хеш должен быть устойчивым к атакам перебора паролей.

2.1.2.2. Используемая информация

Для алгоритма регистрации используются:

1. номер зачетной книжки студента или идентификатор преподавателя в системе;
2. уникальный логин пользователя;
3. пароль пользователя.

2.1.2.3. Результаты решения

В результате выполнения алгоритма формируется следующая информация:

* оповещение клиентского приложения об успешной или неудачной регистрации;
* сохранение регистрационных данных пользователя (логин, пароль в виде хеша, идентификатор пользователя) в базе данных.

2.1.2.4. Алгоритм решения

Схема алгоритма представлена на рисунке 1:



Рис. 1

Описание алгоритма:

1. сервер авторизации получает регистрационные данные от клиентского приложения;
2. сервер авторизации сопоставляет полученный номер зачетной книжки или идентификатор преподавателя с данными в базе;
3. если такого студента или преподавателя не существует, то сервер посылает на клиентское приложение ответ с сообщением об ошибке регистрации и алгоритм завершается;
4. если такой студент или преподаватель существует в университете, то алгоритм выполняется дальше;
5. сервер авторизации генерирует «соль» для хеширования пароля – случайный текст, который добавится к введенному паролю для обеспечения уникальности хешей при введении пользователями одинаковых паролей;
6. сервер генерирует хеш пароля и добавлением «соли»;
7. сервер сохраняет регистрационные данные в базу данных пользователей системы;
8. сервер посылает на клиент ответ с сообщением об успешной регистрации и алгоритм завершается.

2.1.3. Описание алгоритма аутентификации и авторизации пользователей

2.1.3.1. Назначение и характеристика

Аутентификация необходима для подтверждения идентичности пользователя. Авторизация необходима для предоставления пользователю прав доступа на необходимые ресурсы.

Алгоритм должен сверять данные пользователя (логин и пароль), пришедшие с клиентского приложения с данными в базе данных, и если все указано верно, то алгоритм должен предоставить доступ пользователю клиентского приложения с помощью токена доступа к необходимым для него ресурсам. Время действия токена должно быть ограничено, для обеспечения безопасности. Алгоритм также должен предоставить метод обновления токена доступа с помощью токена восстановления доступа, для того чтобы токен доступа можно было восстановить без введения логина и пароля пользователя. В добавок к токенам, алгоритм должен отправлять на клиентское приложение идентификатор пользователя в системе.

2.1.3.2. Используемая информация

При первой аутентификации используются:

* логин пользователя;
* пароль пользователя.

При наличии токена доступа для аутентификации и авторизации используется этот токен доступа.

При обновлении токена доступа используются:

* токен доступа;
* токен восстановления доступа.

2.1.3.3. Результаты решения

В результате выполнения алгоритма при первой аутентификации и при обновлении токена доступа формируется следующая информация:

* токен доступа;
* токен восстановления доступа;
* идентификатор пользователя в системе.

Токен восстановления доступа сохраняется в базе данных пользователей. Все сформированные данные отправляются на клиентское приложение.

При наличии актуального токена – пользователю предоставляется доступ к необходимым ресурсам.

2.1.3.4. Алгоритм решения

Схема алгоритма при наличии токена доступа не представлена, этот вариант рассмотрен в текстовом описании алгоритма. Схемы алгоритма при первой аутентификации и при обновлении токена доступа представлены на рисунках 2 и 3, соответственно.

|  |  |
| --- | --- |
|  |  |
| Рис. 2 | Рис. 3 |

Описание алгоритма при первой аутентификации пользователя:

1. сервер получает данные для аутентификации с клиентского приложения (логин и пароль);
2. сервер сверяет введенные данные с данными из БД, пароль сверяется с помощью вычисление хеша как в алгоритме регистрации пользователей;
3. если данные введены неверно, то на клиентское приложение возвращается ошибка;
4. если данные введены верно, то сервер генерирует токен доступа и токен восстановления доступа;
5. сервер сохраняет токен восстановления доступа в базе данных;
6. сервер отправляет на клиент токен доступа, токен восстановления доступа и идентификатор пользователя в системе.

Описание алгоритма при наличии токена доступа:

1. сервер получает токен доступа;
2. сервер проверяет валидность полученного токена доступа;
3. если токен не валиден, то на клиентское приложение отправляется ошибка авторизации с информацией, показывающей, почему токен не валиден (если время действия токена истекло);
4. если токен валиден, то сервер предоставляет пользователю доступ к запрашиваемому ресурсу.

Описание алгоритма при восстановлении токена доступа:

1. сервер получает токен доступа и токен восстановления доступа;
2. сервер проверяет валидность токена доступа (опуская проверку времени действия токена);
3. если токен доступа не валиден, то сервер отправляет на клиентское приложение ошибку с сообщением о неправильном токене доступа;
4. сервер проверяет валидность токена восстановления доступа;
5. если токен восстановления доступа не валиден, то сервер отправляет на клиентское приложение ошибку с сообщением о неправильном токене восстановления доступа;
6. если полученные токены валидны, то сервер генерирует токен доступа и токен восстановления доступа;
7. сервер сохраняет токен восстановления доступа в базе данных;
8. сервер отправляет на клиент токен доступа, токен восстановления доступа и идентификатор пользователя в системе.

2.2. Разработка сервиса ресурсов

2.2.1. Описание постановки задачи

2.2.1.1. Характеристика задачи

Задача – разработка сервиса ресурсов.

Сервис ресурсов разрабатывается с целью предоставления клиентским приложениям для студентов и преподавателей интерфейса взаимодействия с базой данных системы. Для реализации возможности масштабирования системы в будущем на большое количество университетов – было решено спроектировать собственную базу данных и разработать сервис ресурсов, который будет предоставлять доступ (API) мобильным приложениям пользователей к данным системы.

Разработка сервиса ресурсов связана с разработкой сервиса авторизации, так как через сервис авторизации пользователям будет предоставляться доступ к методам сервера ресурсов.

Разработка задачи будет завершена при успешном прохождении тестирования сервера.

2.2.1.2. Входная информация

Вся входная информация приходит в составе http запросов.

2.2.1.2.1. Входная информация студенческого приложения

Входная информация при запросе расписания:

* идентификатор группы студента пользователя;
* дата, за которую необходимо выдать расписание.

Входная информация при запросе информации о предмете:

* идентификатор предмета.

Входная информация при запросе списка студентов группы:

* идентификатор группы студента пользователя.

Входная информация при запросе списка изучаемых предметов группы:

* идентификатор студента пользователя.

Входная информация при запросе информации о преподавателе:

* идентификатор преподавателя.

Входная информация при запросе личного профиля:

* идентификатор студента пользователя.

Входная информация при редактировании личного профиля:

* идентификатор пользователя студента;
* редактируемая информация (почтовый адрес или номер телефона).

Входная информация при запросе новостей и объявлений:

* идентификатор группы студента пользователя.

Входная информация при опубликовании объявления:

* идентификатор пользователя студента.

2.2.1.2.2. Входная информация преподавательского приложения

Входная информация при запросе расписания:

* идентификатор преподавателя пользователя;
* дата, за которую необходимо выдать расписание.

Входная информация при запросе информации о предмете:

* идентификатор предмета.

Входная информация при запросе списка обучаемых групп:

* идентификатор преподавателя пользователя.

Входная информация при запросе списка студентов одной из обучаемых групп:

* идентификатор группы студентов.

Входная информация при запросе списка преподаваемых предметов в группе:

* идентификатор преподавателя пользователя;
* идентификатор группы студентов.

Входная информация при запросе информации о студенте:

* идентификатор студента.

Входная информация при запросе личного профиля:

* идентификатор преподавателя пользователя.

Входная информация при редактировании личного профиля:

* идентификатор пользователя преподавателя;
* редактируемая информация (почтовый адрес или номер телефона).

Входная информация при запросе новостей и объявлений:

* идентификатор преподавателя пользователя.

Входная информация при опубликовании объявления:

* идентификатор пользователя преподавателя.

2.2.1.3. Выходная информация

Вся выходная информация отправляется в качестве ответа на http запрос. Для получения все данных – необходим токе доступа, который формируется сервисом авторизации.

2.2.1.3.1. Выходная информация студенческого приложения

Выходная информация при запросе расписания:

* расписание на запрошенный день в виде списка предметов с датами.

Выходная информация при запросе информации о предмете:

* данные по предмету (описание, преподаватель).

Выходная информация при запросе списка студентов группы:

* список студентов группы.

Выходная информация при запросе списка изучаемых предметов группы:

* список изучаемых предметов группы.

Выходная информация при запросе информации о преподавателе:

* информация о преподавателе.

Выходная информация при запросе личного профиля:

* информация из личного профиля студента пользователя.

Выходная информация при редактировании личного профиля:

* сообщение об успешном редактировании.

Выходная информация при запросе новостей и объявлений:

* список новостей и объявлений.

Выходная информация при опубликовании объявления:

* сообщение об успешной публикации.

2.2.1.3.2. Выходная информация преподавательского приложения

Выходная информация при запросе расписания:

* расписание в виде списка предметов с датами.

Выходная информация при запросе информации о предмете:

* данные по предмету (описание, преподаватель).

Выходная информация при запросе списка обучаемых групп:

* список обучаемых групп.

Выходная информация при запросе списка студентов одной из обучаемых групп:

* список студентов группы.

Выходная информация при запросе списка преподаваемых предметов в группе:

* список преподаваемых предметов в группе.

Выходная информация при запросе данных о студенте:

* данные о студенте.

Выходная информация при запросе личного профиля:

* информация из личного профиля преподавателя.

Выходная информация при редактировании личного профиля:

* сообщение об успешном редактировании.

Выходная информация при запросе новостей и объявлений:

* список новостей и объявлений.

Выходная информация при опубликовании объявления:

* сообщение об успешной публикации.

2.2.2. Описание алгоритма предоставления данных для приложений преподавателя и студента

2.2.2.1. Назначение и характеристика

Алгоритм предоставления данных для приложений преподавателя и студента необходим для сбора и формирования в удобный вид данных из БД, с последующей отправкой их на клиентское приложение.

Алгоритм должен вытаскивать данные для приложения преподавателя из БД, при этом формируя их в тот вид, в котором они необходимы в клиентском приложении, чтобы как можно больше уменьшить количество бизнес-логики на клиенте, что поспособствует увеличению производительности. Алгоритм должен предоставлять доступ к своим функциям только при наличии валидного токена доступа, который можно получить, используя сервис авторизации.

2.2.2.2. Используемая информация

Для авторизации пользователя используется токен доступа.

Информация, которая используется только для приложения преподавателя или только для приложения студента – подписана соответствующим образом. Информация, используемая в обоих случаях – не подписана.

При выполнении запроса на выдачу расписания используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор группы студента-пользователя (приложение студента)
* данные таблицы расписания занятий из БД;
* данные таблицы студенческих групп из БД (приложение преподавателя);
* данные таблицы преподавателей из БД (приложение студента);
* данные таблицы предметов из БД.

При выполнении запроса на выдачу данных об определенном предмете используется следующая информация:

* идентификатор этого предмета;
* данные таблицы предметов из БД.

При выполнении запроса на выдачу списка обучаемых преподавателем студенческих групп используется следующая информация (приложение преподавателя):

* идентификатор преподавателя, выполняющего запрос;
* данные таблицы, связывающей студенческие группы и преподавателей из БД.

При выполнении запроса на выдачу списка студентов одной из групп используется следующая информация:

* идентификатор требуемой группы;
* данные таблицы студенческих групп из БД.

При выполнении запроса на выдачу списка преподаваемых предметов в определенной группе используется следующая информация (приложение преподавателя):

* идентификатор группы;
* идентификатор преподавателя, выполняющего запрос;
* данные таблицы предметов из БД;
* данные таблицы связывающей предметы и группы из БД.

При выполнении запроса на выдачу данных о студенте используется следующая информация:

* идентификатор требуемого студента;
* данные таблицы реквизитов студентов из БД.

При выполнении запроса на выдачу данных о преподавателе используется следующая информация:

* идентификатор требуемого преподавателя;
* данные таблицы реквизитов сотрудников университета из БД.

При выполнении запроса на редактирование данных из личного профиля используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* редактируемые данные;
* данные таблицы реквизитов сотрудников университета из БД (приложение преподавателя);
* данные из таблицы реквизитов студентов (приложение студента);

При выполнении запроса на выдачу новостей и объявлений используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* данные таблицы новостей из БД;
* данные таблицы пользовательских объявлений из БД.

При выполнении запроса на публикацию объявления используется следующая информация:

* идентификатор преподавателя-пользователя (приложение преподавателя);
* идентификатор студента-пользователя (приложение студента);
* идентификаторы групп, для которых публикуется объявление (приложение преподавателя);
* текст объявления;
* данные таблицы пользовательских объявлений из БД.

При выполнении запроса на выдачу списка изучаемых студентом предметов используется следующая информация (приложение студента):

* идентификатор студента-пользователя.

2.2.2.3. Результаты решения

Информация, которая формируется только для приложения преподавателя или только для приложения студента – подписана соответствующим образом. Информация, формируемая в обоих случаях – не подписана.

В результате выполнения запроса на выдачу расписания формируется список предметов. Объект предмета для расписания включает в себя:

* идентификатор предмета;
* дата и время проведения пары;
* наименование студенческой группы, в которой будет проводиться пара (приложение преподавателя);
* ФИО преподавателя (приложение студента).

В результате выполнения запроса на выдачу данных об определенном предмете формируется объект предмета, в составе которого:

* идентификатор предмета;
* описание предмета;
* идентификатор преподавателя (приложение студента);
* ФИО преподавателя (приложение студента).

В результате выполнения запроса на выдачу списка обучаемых преподавателем студенческих групп формируется список групп (приложение преподавателя). Объект группы включает в себя:

* идентификатор группы;
* наименование группы.

В результате выполнения запроса на выдачу списка студентов одной из групп, формируется список студентов. Объект студента включает в себя:

* идентификатор студента;
* ФИО студента.

В результате выполнения запроса на выдачу данных о студенте формируется объект студента, в составе которого:

* идентификатор студента;
* ФИО студента;
* почтовый адрес студента;
* номер телефона студента;
* фото студента.

В результате выполнения запроса на выдачу данных о преподавателе формируется объект преподавателя, в составе которого:

* идентификатор преподавателя;
* ФИО преподавателя;
* номер телефона преподавателя;
* текущий статус преподавателя;
* должность преподавателя.

В результате запроса на редактирование данных из личного профиля пользователя формируется сообщение об успешном редактировании.

В результате выполнения запроса на публикацию объявления формируется сообщение об успешной публикации.

2.2.2.4. Алгоритм решения

Схема алгоритма представлена на рисунке 4.



Рис. 4

Описание алгоритма:

1. при поступлении запроса происходит определение метода, который его выполнит (маршрутизация запроса);
2. на данный момент ко всем методам сервиса ресурсов отсутствует анонимный доступ, поэтому перед выполнением запроса происходит валидация токена доступа пользователя с помощью сервиса авторизации;
3. если токен доступа пользователя не валиден, то сервис ресурсов отказывает пользователю в доступе;
4. если токен валиден, то сервис ресурсов выполняет необходимый запрос.

2.3. Описание контрольного примера

2.3.1. Назначение

Контрольным примером проверяются:

* регистрация пользователя в системе;
* занесение нового пользователя в базу;
* хеширование пароля.
* выдача токена доступа и токена восстановления доступа пользователю;
* выдача токенов при первой аутентификации;
* выдача токенов при восстановлении доступа;
* сохранение токена восстановления в базе.
* доступ к методам сервиса ресурсов;
* доступ при валидном токене доступа;
* доступ при не валидном токене доступа;
* работа метода по предоставлению данных о студенте.

2.3.2. Исходные данные

Перечень исходных данных:

* запись в базе о группе Б08-191-2 c идентификатором 1;
* запись в базе о студенте Иванове Иване Ивановиче с номером зачетной книжки 123 и идентификатором группы 1, номером мобильного телефона 12345, номером домашнего телефона 12345;
* логин, который будет использоваться для регистрации этого студента в системе IvanovII;
* пароль, который будет использоваться для регистрации этого студента в системе qwerty.

2.3.4. Результаты расчета

Результаты проверки регистрации пользователя представлены на рисунках 5 и 6.

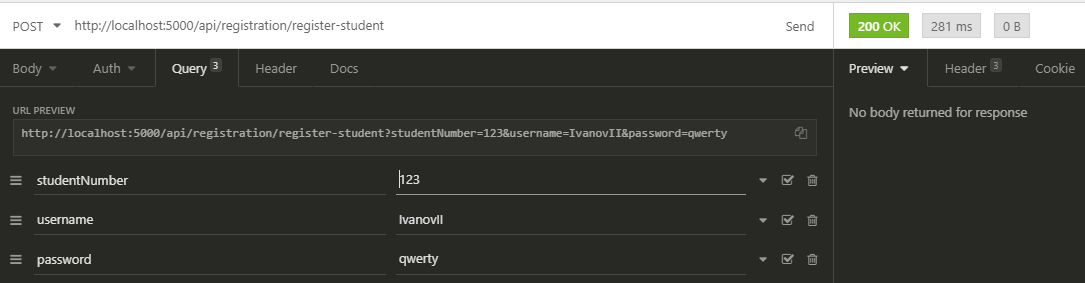


Рис. 5. Пример работы регистрации



Рис. 6. Пример занесения зарегистрированного пользователя в базу

Результаты проверки выдачи токена доступа пользователю представлены на рисунках 7, 8 и 9.

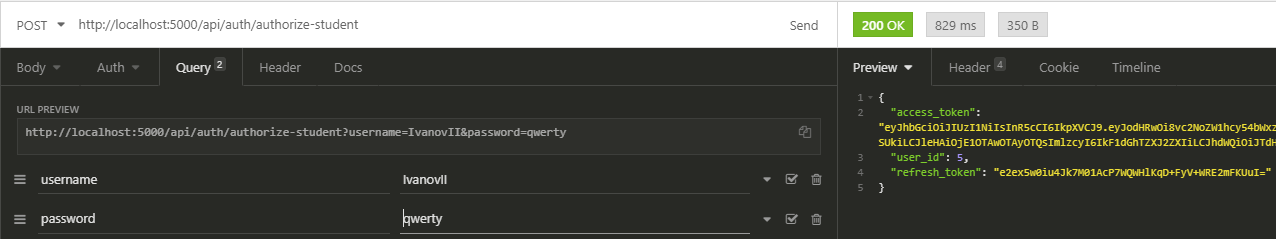


Рис. 7. Пример первой аутентификации

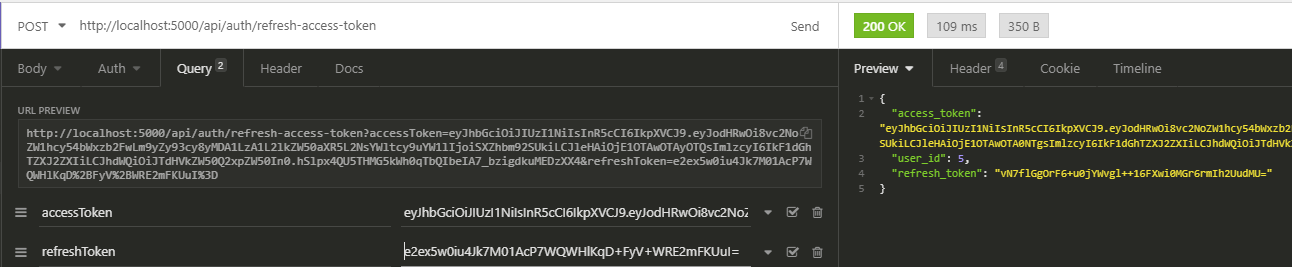


Рис. 8. Пример восстановления токена доступа



Рис. 9. Пример сохранения токена восстановления в базе

Результаты проверки доступа к методам сервиса ресурсов представлены на рисунках 10, 11, 12.

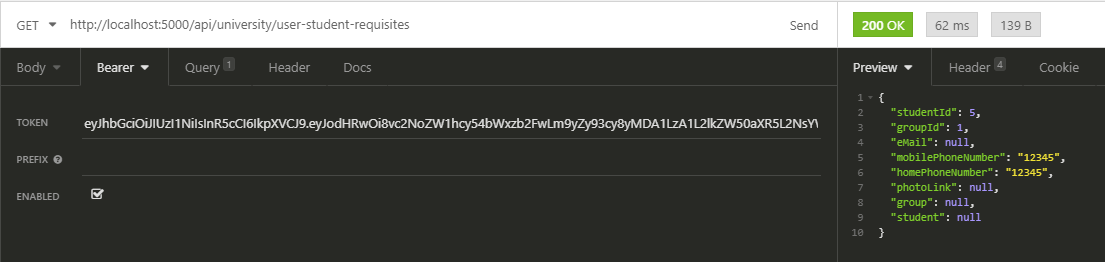


Рис. 10. Пример доступа к методу предоставления данных о студенте

с помощью валидного токена доступа

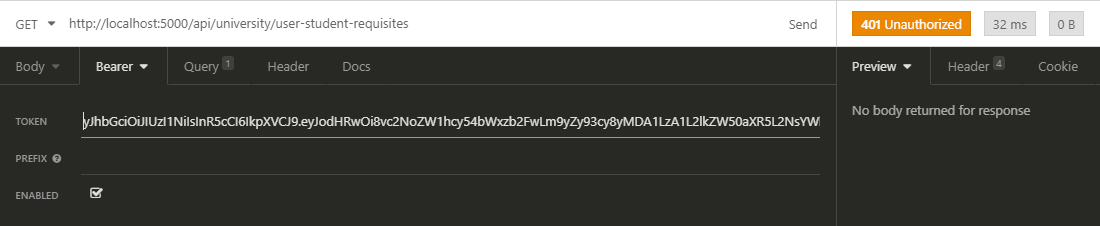


Рис. 11. Пример доступа к методу предоставления данных о студенте

с помощью не валидного токена доступа

2.3.5. Результаты испытания программы

Для проверки работы сервисов использовались: программа для тестирования API – Insomnia, среда разработки СУБД PostgreSQL – dbForge for PostgreSQL, среда разработки платформы .NET Core – Visual Studio 2019.

По результатам проверки удалось выяснить, что разработанная часть системы функционирует безошибочно.